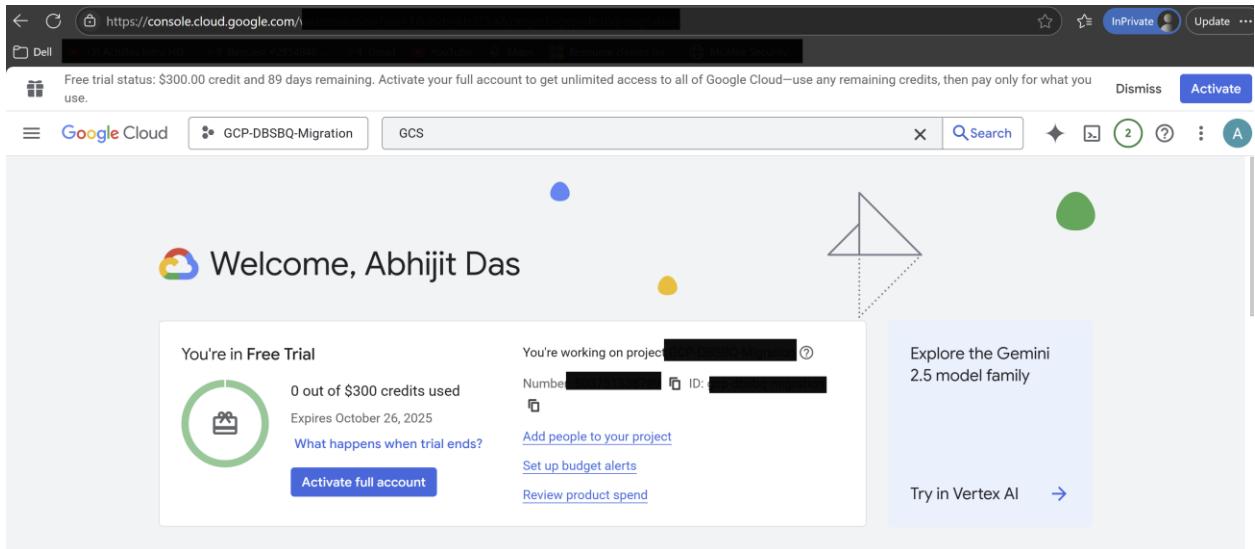


Pre-requisites

1. Your data is already in GCS (e.g., CSV, JSON, or Parquet format).
2. You have a **BigQuery dataset** created (e.g., `my_dataset`).
3. The BigQuery service account has **read access to the GCS bucket**.

Create Google Cloud Account:



Create buckets

The screenshot shows a search results page for "GCS". The results include:

- Cloud Storage**: Enterprise-ready object storage
- Storage Transfer**: Secure and flexible way to move data
- Google Cloud: Cloud Computing Services**: Meet your business challenges head on with cloud computing...
- SFTP Gateway Enterprise**: Thorn Technologies LLC
- Cloud Storage**: Cloud Storage is a managed service for storing unstructured dat...
- Storage classes**: This page explains the concept of storage class and the...
- Infinia**: DDN
- Cloud Storage documentation**: Cloud Storage allows world-wide storage and retrieval of any...
- Product overview of Cloud Storage**: Cloud Storage is a service for storing your objects in Google...
- WEKA® Data Platform**: Weka IO

Create GCS bucket and upload files to folders in buckets

The screenshot shows the Google Cloud Storage browser interface. The left sidebar shows:

- Cloud Storage
- Buckets (selected)
- Monitoring
- Settings
- Storage Intelligence
- Insights datasets
- Configuration
- Marketplace
- Release Notes

The main area shows "Bucket details" for "retail-demo-data-abhijit". The "Objects" tab is selected, showing a "Folder browser" with the following structure:

```

    Folder browser
    └── retail-demo-data-abhijit
        ├── customers/
        ├── inventory/
        ├── order_items/
        ├── orders/
        ├── products/
        └── stores/
    
```

Below the folder browser, there is a table of objects:

Name	Type	Created
stores.csv	text/csv	Jul 29, 2025, 12:21:39 PM

A message at the bottom says "1 file successfully uploaded". On the right, there is a "Uploads and operations" section showing:

File	Status
inventory.csv	Complete
order_items.csv	Complete
orders.csv	Complete
stores.csv	Complete
products.csv	Complete

Now we need to update GCS bucket → Permissions to access BigQuery Data - BigQuery Data Editor / BigQuery Job User

0.00 credit and 91 days remaining. Activate your full account to get unlimited storage and processing power.

Create table

Source

Create table from Google Cloud Storage

Select file from GCS bucket or use a URI pattern * retail-demo-data-abhijit/inventory/inventory.csv [Browse](#)

File format CSV

Source Data Partitioning

Destination

Project * gcp-dbsbq-migration [Browse](#)

Dataset * retails_db

Table * inventory_tbl

Maximum name size is 1,024 UTF-8 bytes. Unicode letters, marks, numbers, connectors, dashes, and spaces are allowed.

Table type External table

Regional / dual region GCS buckets are recommended for External table.

Create a BigLake table using a Cloud resource connection

Schema

Auto detect

Edit as text

[Create table](#) [Cancel](#)

inventory_tbl [Query](#) [Open in](#) [Share](#) [Delete](#)

[Schema](#) [Details](#) [Insights](#) [Lineage](#) [Data Profile](#) [Data Quality](#)

Table info

[Edit Details](#)

Table ID	retails_db.inventory_tbl
Created	Jul 29, 2025, 11:34:18 PM UTC-7
Last modified	Jul 29, 2025, 11:34:18 PM UTC-7
Table expiration	NEVER
Data location	US-west1
Case insensitive	false
Description	
Labels	
Primary key(s)	
Tags	

External Data Configuration

Source URI(s)	gs://[REDACTED]/inventory/inventory.csv
Auto-detect schema	false
Ignore unknown values	false
Source format	CSV
Max bad records	0
Allow jagged rows (CSV)	false
Allow quoted newlines (CSV)	false
Field delimiter (CSV)	,

Load remaining files to BigQuery tables.

The screenshot shows the BigQuery interface with a search bar at the top. A checkbox labeled "Show starred only" is checked. Below the search bar is a list of resources:

- Queries
- Notebooks
- Data canvases
- Data preparations
- Pipelines
- External connections
- customers_tbl**
- retails_db**
 - customer_tbl**
 - inventory_tbl**
 - order_items_tbl**
 - orders_tbl**
 - products_tbl**
 - stores_tbl**

A hand-drawn curly brace is drawn around the "retails_db" dataset and its contained tables.

Integration with Composer (GCP) and BigQuery

Create Composer environment

The screenshot shows the Google Cloud Platform interface for creating a Cloud Composer environment. The page title is "Create a Composer 3 environment". The form fields include:

- Name:
- Location:
- Image version:
- Service account:
- We recommend you to set up a user-managed service account for Cloud Composer environments. Learn more about environment's service account.
- Labels:
- Resilience mode: Standard resilience
- Failure database zone:

Environments													
	Actions												
	Actions												
Create	Refresh	Delete											
<p>Join Airflow community on October 7th - 9th during the Airflow Summit 2025 conference to learn more about Airflow and share your expertise. Register here</p>													
Filter	Filter environments												
State	Name ↑	Location	Composer version	Airflow version	Creation time	Update time	Airflow webserver	DAG list	Logs	DAGs folder	Labels		
Running	bg-pipeline-demo	us-west1	3	2.10.5-build.10	7/30/25, 9:44 AM	7/30/25, 9:44 AM	None	DAGs	Logs	None	None		

Grant IAM Permissions to Composer's Service Account

Composer / Environment: bq-pipeline-demo / Configuration	
Environment details Open Airflow UI Open DAGs folder Save snapshot Load snapshot Refresh Delete	
<p>ⓘ While this environment is being created, it cannot be edited or deleted.</p>	
 bq-pipeline-demo	This environment is being created
Monitoring	Logs
DAGs	Environment configuration
Airflow configuration overrides	Environment variables
Labels	Pypi packages
Name	bq-pipeline-demo
Location	us-west1
Service account	[REDACTED] compute@developer.gserviceaccount.com
Image version	composer-3-airflow-2.10.5-build.10 Upgrade ⓘ Newest available version
Python version	3
DAGs folder	None
Airflow web UI	—
Logging	view logs
Maintenance windows	Default Edit
Database data retention policy	60 days New Edit
Dataplex data lineage integration	Disabled Edit
Data encryption key	Google-managed
Resilience mode	Standard resilience Edit
Web server plugins	Enabled Edit
Created	Wed Jul 30 2025 09:44:54 GMT-0700 (Pacific Daylight Time)
Updated	Wed Jul 30 2025 09:44:54 GMT-0700 (Pacific Daylight Time)
<hr/>	
Resources	
Workloads configuration	Edit
Scheduler	1 scheduler with 0.5 vCPU, 2 GB memory, 1 GB storage
DAG processor	1 DAG processor with 1 vCPU, 4 GB memory, 1 GB storage
Triggerer	1 triggerer with 0.5 vCPU, 1 GB memory, 1 GB storage
Web server	1 vCPU, 2 GB memory, 1 GB storage

Free trial status: \$300.00 credit and 89 days remaining. Activate your full account to get unlimited access to all of Google Cloud.

Google Cloud [REDACTED] migration iam & admin

IAM & Admin / IAM

IAM

Allow Deny Recommendations history

Permissions for project [REDACTED] migration

These permissions affect this project and all of its resources. [Learn more](#)

View by principals View by roles

+ Grant access - Remove access

Filter Enter property name or value

Type	Principal
[REDACTED]	[REDACTED] compute@developer.gserviceaccount.com
[REDACTED]	[REDACTED] abhijit@gmail.com

Principal [REDACTED] compute@developer.gserviceaccount.com Project [REDACTED] migration

Assign roles

Roles are composed of sets of permissions and determine what the principal can do with this resource. [Learn more](#)

Role Editor IAM condition (optional) [REDACTED] + Add IAM condition

View, create, update, and delete most Google Cloud resources. See the list of included permissions.

+ Add another role Help me choose roles

Save Test changes Cancel

The screenshot shows the Google Cloud IAM (Identity and Access Management) interface for a specific project. The main area displays a list of principals (users and service accounts) and the roles they have been granted. In this case, the principal 'compute@developer.gserviceaccount.com' has been assigned the 'Editor' role. The interface also includes tabs for 'View by principals' and 'View by roles', and a 'Grant access' section. A sidebar on the left provides navigation links for various Google Cloud services, such as PAM, Principal Access Boundaries, and Workload Identity Federation. The top of the screen shows a free trial status with \$300.00 credit and 89 days remaining.

Edit access to "GCP-DBSBQ-Migration"

Principal [?](#)

[REDACTED]compute@developer.gserviceaccount.com

Project

[REDACTED]migration

Assign roles

Roles are composed of sets of permissions and determine what the principal can do with this resource. [Learn more](#)

Role

Editor

IAM condition (optional) [?](#)

[+ Add IAM condition](#)



View, create, update, and delete most Google Cloud resources. See the list of included permissions.

Role

BigQuery Job User

IAM condition (optional) [?](#)

[+ Add IAM condition](#)



Access to run jobs

Role

BigQuery Admin

IAM condition (optional) [?](#)

[+ Add IAM condition](#)



Administer all BigQuery resources and data

Role

Storage Object Viewer

IAM condition (optional) [?](#)

[+ Add IAM condition](#)



Grants access to view objects and their metadata, excluding ACLs. Can also list the objects in a bucket.

Role

BigQuery Data Editor

IAM condition (optional) [?](#)

[+ Add IAM condition](#)



Access to edit all the contents of datasets

[+ Add another role](#)

Help me choose roles

[Save](#)

[Test changes](#)

[?](#)

[Cancel](#)

Create bucket and upload sql files which we need to run by using Airflow DAGs

The screenshot shows the Google Cloud Storage interface. On the left, there's a sidebar with 'Cloud Storage' selected. The main area shows a bucket named 'retails-biggquery-sqls'. Under 'Objects', there's a folder browser showing a single file named 'bronze_queries.sql' located in a folder 'bqsqls/'. The file is 1 KB in size, has a type of 'application/octet-stream', and was created on Aug 1, 2025, at 7:31:01 AM. It has 'Standard' storage class and 'Not public' public access.

Script To run (DAG)

```
from airflow import models
from airflow.providers.google.cloud.operators.bigquery import BigQueryInsertJobOperator
from airflow.utils.dates import days_ago

PROJECT_ID = "your-project-id"
BUCKET_NAME = "retails-[REDACTED]"
SQL_FILE = "[REDACTED]bronze_queries.sql"
DATASET = "retail_bronze"
LOCATION = "US" # or EU, asia-east1, etc.

with models.DAG(
    dag_id="trigger_bq_sql_from_gcs",
    schedule_interval=None, # Trigger manually or via another DAG
    start_date=days_ago(1),
    catchup=False,
    tags=["bigquery", "gcs"],
) as dag:

    bq_query_from_gcs = BigQueryInsertJobOperator(
        task_id="run_bq_sql_from_gcs",
        configuration={
            "query": {
                "query": f"gs://{BUCKET_NAME}/{SQL_FILE}",
                "useLegacySql": False,
            }
        },
        location=LOCATION,
        project_id=PROJECT_ID,
    )
```

Or we can hard-code sql queries in DAG python script.

```
1  from airflow import models
2  from airflow.providers.google.cloud.operators.bigquery import BigQueryInsertJobOperator
3  from airflow.utils.dates import days_ago
4
5  PROJECT_ID = "gcp-dbsbq-migration"
6  LOCATION = "us-west1"
7
8  # SQL for Bronze
9  SQL_QUERY_Bronze = """
10 --Create bronze tables with schema
11 CREATE SCHEMA IF NOT EXISTS `gcp-dbsbq-migration.retail_bronze`;
12
13
14 -- customers
15 CREATE OR REPLACE TABLE `gcp-dbsbq-migration.retail_bronze.customers_tbl`
16 AS SELECT * FROM `gcp-dbsbq-migration.retails_db.customer_tbl`;
17
18 --inventory
19 CREATE OR REPLACE TABLE `gcp-dbsbq-migration.retail_bronze.inventory_tbl`
20 AS SELECT * FROM `gcp-dbsbq-migration.retails_db.inventory_tbl`;
21
22 --order_items
23 CREATE OR REPLACE TABLE `gcp-dbsbq-migration.retail_bronze.order_items_tbl`
24 AS SELECT * FROM `gcp-dbsbq-migration.retails_db.order_items_tbl`;
25
26 --orders
27 CREATE OR REPLACE TABLE `gcp-dbsbq-migration.retail_bronze.orders_tbl`
28 AS SELECT * FROM `gcp-dbsbq-migration.retails_db.orders_tbl`;
29
```

```

) as dag:
    silver_task = BigQueryInsertJobOperator(
        configuration={
            },
            location=LOCATION,
            project_id=PROJECT_ID,
        )

    gold_task = BigQueryInsertJobOperator(
        task_id="process_gold",
        configuration={
            "query": {
                "query": SQL_QUERY_Gold,
                "useLegacySql": False
            }
        },
        location=LOCATION,
        project_id=PROJECT_ID,
    )

    ml_model_task = BigQueryInsertJobOperator(
        task_id="build_model",
        configuration={
            "query": {
                "query": SQL_QUERY_MLModel,
                "useLegacySql": False
            }
        },
        location=LOCATION,
        project_id=PROJECT_ID,
    )

# Set execution order: Bronze -> Silver -> Gold
bronze_task >> silver_task >> gold_task >> ml_model_task

```

Upload python scripts (DAGs) in Airflow -> Composer -> DAG Folder

The screenshot shows the Google Cloud Storage interface. On the left, the navigation pane includes 'Cloud Storage' (selected), 'Overview', 'Buckets', 'Monitoring', 'Settings', 'Storage Intelligence', 'Insights datasets', and 'Configuration'. The main area displays a bucket named 'us-west1-bq-pipeline' (redacted). Under 'Buckets', it shows 'Location: us-west1 (Oregon)', 'Storage class: Standard', 'Public access: Subject to object ACLs', and 'Protection: Soft Delete'. Below this, the 'Objects' tab is selected, showing a 'Folder browser' for a folder named 'us-west1-bq-pipeline-demo' (redacted). This folder contains sub-folders 'dags/' and 'data/' and files 'airflow_monitoring.py' and 'bronze_queries.sql'. A search bar at the top right contains 'Search (/) for resources, docs, products, and more' and a 'Search' button.

Name	Type	Created	Storage class	Last modified	Public access
airflow_monitoring.py	text/x-python	Jul 30, 2025, 10:03:14 AM	Standard	Jul 30, 2025, 10:03:14 AM	Not public
bronze_queries.sql	application/octet-stream	Aug 1, 2025, 7:37:49 AM	Standard	Aug 1, 2025, 7:37:49 AM	Not public

bq-pipeline-demo

All 2 Active 2 Paused 0

Running 1 Failed 0

Filter DAGs by tag

Search DAGs

DAG Owner Runs Schedule Last Run Next Run Recent Tasks Actions Links

airflow_monitoring airflow 2025-08-01, 17:50:00 2025-08-01, 18:00:00 ...

run_retails_data_sqls airflow 2025-08-01, 18:01:16 None ...

Showing 1-2 of 2 DAGs

DAG: run_retails_data_sqls

08/02/2025 06:35:37 AM

All Run Types All Run States Clear Filters

Press shift + / for Shortcuts

Duration

00:02:21

00:01:10

00:00:00

process_bronze process_silver process_gold build_model

DAG run_retails_data_sqls

Details Graph Gantt Code Event Log Run Duration Task Duration Calendar

process_bronze BigQueryInsertJobOperator

process_silver BigQueryInsertJobOperator

process_gold BigQueryInsertJobOperator

build_model BigQueryInsertJobOperator

Build and run ML model on BigQuery

```
1 --Create ML Models
2
3 CREATE OR REPLACE MODEL `████████migration.retail_gold.ml_churn_model`
4 OPTIONS(model_type='logistic_reg') AS
5 SELECT
6   *,
7   CASE WHEN days_since_last_order > 60 THEN 1 ELSE 0 END AS label
8 FROM `████████migration.retail_gold.gold_customer_features`;
9
10
11 SELECT
12   customer_id,
13   predicted_label,
14   predicted_label_probs
15 FROM
16   ML.PREDICT(MODEL `████████migration.retail_gold.ml_churn_model`,
17   (SELECT * FROM `████████migration.retail_gold.gold_customer_features`));
```

▶ Query completed

Query results

 Sav

Job information		Results	Chart	JSON	Execution details	Execution graph	
Row	customer_id			predicted_label	predicted_label_probs	label	predicted_label_p...
1	d3dec6a7-5d04-4b04-a732-3fc3cd961148			1		1	0.999999872247...
						0	1.277521007869...
2	f76ba400-93fa-46a6-b91a-927cf0e05ef8			1		1	0.999999889427...
						0	1.105724314331...
3	fab12427-b911-46f5-b230-d659a54c464e			1		1	0.999999727783...
						0	2.722164066382...
4	b3209e23-fcdc-4af0-9a12-ebfe383b5dbe			0		1	6.857382860765...

To capture the cost of BigQuery sqls which are triggered by Airflow, we need to find the job_id from the Airflow logs.

DAG: run.retails_data_sqls Run: 2025-08-01, 18:01:16 UTC Task: process_bronze

Logs:

```
-- customers
CREATE OR REPLACE TABLE [REDACTED] AS SELECT * FROM [REDACTED].retail_bronze.customers_tbl\` AS SELECT * FROM [REDACTED].migration_retail_bronze.customer_tbl\` AS SELECT * FROM [REDACTED].retail_bronze.inventory_tbl\` AS SELECT * FROM [REDACTED].migration_retail_bronze.inventory_tbl\` AS SELECT * FROM [REDACTED].retail_bronze.order_items_tbl\` AS SELECT * FROM [REDACTED].migration_retail_bronze.order_items_tbl\` AS SELECT * FROM [REDACTED].retail_bronze.orders_tbl\` AS SELECT * FROM [REDACTED].migration_retail_bronze.orders_tbl\` AS SELECT * FROM [REDACTED].retail_bronze.products_tbl\` AS SELECT * FROM [REDACTED].migration_retail_bronze.products_tbl\` AS SELECT * FROM [REDACTED].retail_bronze.stores_tbl\` AS SELECT * FROM [REDACTED].migration_retail_bronze.stores_tbl\`
```

WARN: [2025-08-01, 18:01:12 UTC] [credential_provider.py:140] INFO - Getting connection using 'google.auth.default()' since no explicit credentials are provided.

WARN: [2025-08-01, 18:01:16 UTC] [bigquery.py:134] INFO - Inserting job airflow-run_retails_data_sqls_process_bronze_2025_08_01T18_01_16_075753 into dataset 0083f9c968e0c is completed. Checking the job status

INFO: [2025-08-01, 18:01:16 UTC] [taskinstance.py:230] INFO - Task exited with return code 0

INFO: [2025-08-01, 18:01:17 UTC] [local_task_job_runner.py:246] INFO - Task waited with return code 0

INFO: [2025-08-01, 18:01:17 UTC] [taskinstance.py:198] INFO - 3 downstream tasks scheduled from follow-on schedule check

INFO: [2025-08-01, 18:01:17 UTC] [local_task_job_runner.py:245] *** Log group end

Then we can run following queries in BigQuery to estimate cost. Below screenshot shows only one job_id cost.

```

1 SELECT *
2 FROM [REDACTED].INFORMATION_SCHEMA.JOBS_BY_PROJECT
3 WHERE job_id = 'airflow_run_retails_data_sqls_process_bronze_2025_08_01T18_01_16_075751_00_00_304a757e5da8290304bd03f0c968e08c';
4
5 SELECT
6   job_id,
7   user_email,
8   creation_time,
9   statement_type,
10  query,
11  total_bytes_processed,
12  total_bytes_processed / POWER(2, 30) AS gb_processed,
13  ROUND(total_bytes_processed / POWER(2, 40) * 5, 2) AS estimated_cost_usd
14  FROM
15  [REDACTED].INFORMATION_SCHEMA.JOBS_BY_PROJECT
16  WHERE
17  job_id = 'airflow_run_retails_data_sqls_process_bronze_2025_08_01T18_01_16_075751_00_00_304a757e5da8290304bd03f0c968e08c'
18  -- AND creation_time BETWEEN TIMESTAMP('2025-08-01 00:00:00') AND TIMESTAMP('2025-08-01 01:00:00')
19  AND user_email = 'compute@developer.gserviceaccount.com'
20 ORDER BY creation_time DESC
21 LIMIT 100;

```

This script will process 1.25 MB when run.

Query results

Job information	Results	Chart	JSON	Execution details	Execution graph																
Row 1	<table border="1"> <tr> <td>Job ID</td> <td>User Email</td> <td>Creation Time</td> <td>Statement Type</td> <td>Query</td> <td>Total Bytes Processed</td> <td>GB Processed</td> <td>Estimated Cost USD</td> </tr> <tr> <td>1 airflow_run_retails_data_sqls... process_bronze_2025_08_01T18_01_16_075751_00_00_304a757e5da8290304bd03f0c968e08c</td> <td>compute@devel...</td> <td>2025-08-01 18:01:23.613000 UTC</td> <td>SCRIPT</td> <td>-Create bronze tables with schema CREATE SCHEMA IF NOT EXISTS</td> <td>505254</td> <td>0.000470554456...</td> <td>0.0</td> </tr> </table>	Job ID	User Email	Creation Time	Statement Type	Query	Total Bytes Processed	GB Processed	Estimated Cost USD	1 airflow_run_retails_data_sqls... process_bronze_2025_08_01T18_01_16_075751_00_00_304a757e5da8290304bd03f0c968e08c	compute@devel...	2025-08-01 18:01:23.613000 UTC	SCRIPT	-Create bronze tables with schema CREATE SCHEMA IF NOT EXISTS	505254	0.000470554456...	0.0				
Job ID	User Email	Creation Time	Statement Type	Query	Total Bytes Processed	GB Processed	Estimated Cost USD														
1 airflow_run_retails_data_sqls... process_bronze_2025_08_01T18_01_16_075751_00_00_304a757e5da8290304bd03f0c968e08c	compute@devel...	2025-08-01 18:01:23.613000 UTC	SCRIPT	-Create bronze tables with schema CREATE SCHEMA IF NOT EXISTS	505254	0.000470554456...	0.0														

Results per page: 50 ▾ 1 – 1 of 1 ▶◀ ▶

Create a separate project for implementing Databricks Project.

New Project

You have 11 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[Manage Quotas](#)

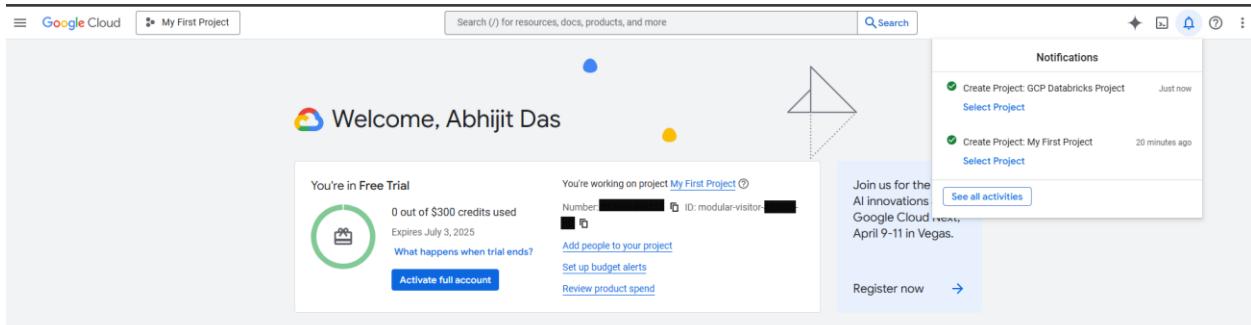
Project name * GCP Databricks Project

Project ID: [REDACTED] It cannot be changed later. [Edit](#)

Location * No organization [Browse](#)

Parent organization or folder

[Create](#) [Cancel](#)



To create GCP Databricks account, we need to subscribe to Databricks.

Find Databricks under Marketplace (Partner Solutions)

The screenshot shows the Google Cloud Marketplace interface. At the top, there's a search bar and a 'My First Project' button. Below the search bar, there are filters for 'Category', 'Type', and 'Price'. The 'Category' filter is set to 'Analytics' (1,117), 'Big data' (584), 'Business Applications' (209), 'Contact Center AI' (13), and 'Databases' (504). The 'Type' filter is set to 'Google Cloud Platform' (36), 'SaaS & APIs' (1,648), 'Virtual machines' (1,919), 'Data' (165), and 'Vertex AI' (23). The 'Price' filter is set to 'Free' (1,117). The main area displays a grid of partner solutions. In the 'Analytics' section, 'Apache Kafka® & Apache Flink® on Confluent Cloud™' by Confluent is highlighted with a '14 Day Free Trial' button. Other solutions shown include FullStory, LiveRamp, Quantum Metric, Shopify, and eCommerce Search, Browse, & Recommendations. In the 'DevOps' section, Datadog, GitLab, Dynatrace SaaS, Snyk, and Harness are listed. A tooltip at the bottom center says 'Now viewing project "My First Project" in organization "No organization"'.

Subscribe

This screenshot is identical to the one above, showing the Google Cloud Marketplace interface for finding Databricks. It displays the same filters, solution grid, and the same tooltip at the bottom center: 'Now viewing project "My First Project" in organization "No organization"'.

≡ Google Cloud

← New Databricks subscription

Pricing Calculator

Order Summary

1. Select Plan

Plan
Databricks

Features

- Managed Apache Spark: Yes
- Optimized Delta Lake: Yes
- Cluster Autopilot: Yes
- Jobs Scheduling & Workflow: Yes
- Notebooks & Collaboration: Yes
- Databricks Runtime for ML: Yes
- Optimized Runtime Engine: Yes
- Administration Console: Yes
- Single Sign-On (SSO): Yes
- Role-based Access Control: Yes
- Token Management API: Yes

Pricing

Usage fee

Databricks Consumption Units <small>?</small>	USD 1.00
/unit	

Free
Estimated total cost

Adjust estimated timeframe

1 day 1 month 1 year

Monthly usage fee

Databricks Consumption Units

Estimated unit 0 unit/mo USD 0.00/mo

2. Purchase details

Select a billing account * —
My Billing Account

3. Terms

Cancellation and change policy

- Your subscription fee is billed every month.

Additional terms

By purchasing, deploying, accessing, or using this product, you acknowledge that Google or an affiliate is the Vendor's agent with respect to this transaction and you agree to comply with the [Google Cloud Marketplace Terms of Service](#) (including the GPC terms set forth in Appendix A of the Marketplace ToS), [Databricks Terms of Service](#) and the terms of applicable open source software licenses bundled with the product.

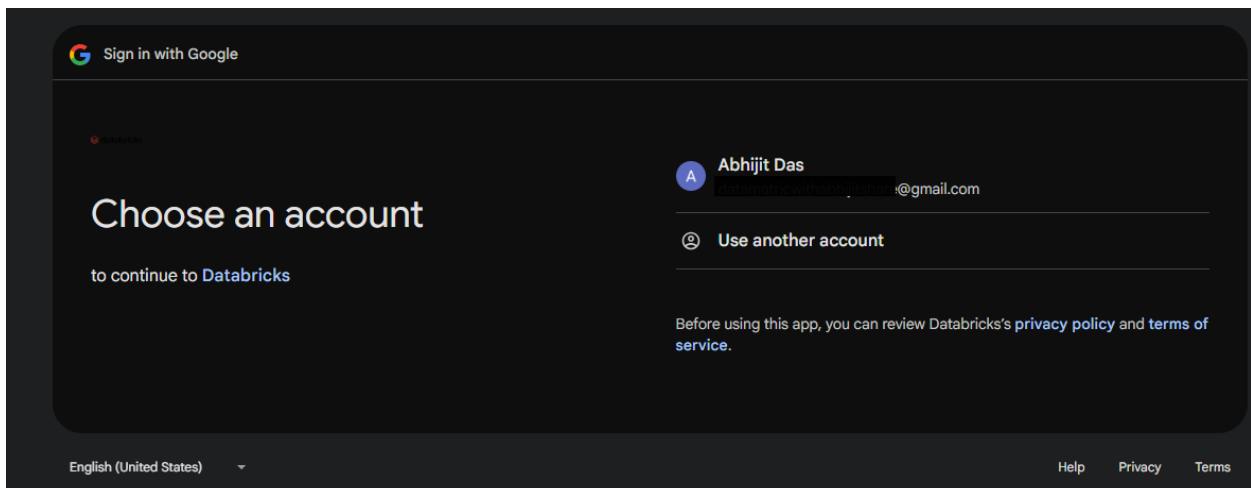
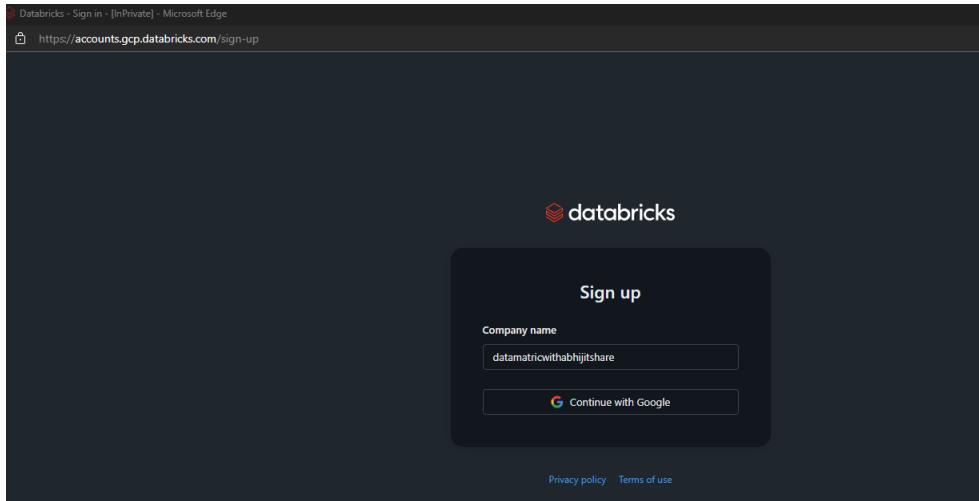
SUBSCRIBE

Your order request has been sent to Databricks



Your new subscription to the Databricks plan for Databricks requires your registration with Databricks. After Databricks approves your request, your subscription will be active and you will begin getting charged. This processing time will depend on the vendor, and you should reach out to Databricks directly with any questions related to signup.

MANAGE ORDERS **SIGN UP WITH DATABRICKS**



Click on Activate

A screenshot of the Google Cloud Billing dashboard. The top navigation bar includes "Activate" and "Dismiss" buttons. The main content area shows a table of billing orders. One row is highlighted with a green checkmark in the "Status" column, indicating it is active. The table columns include: Status, Order number, Order title, Provider, Product, Plan, Next plan, Auto-renew, Purchase date, Start date, End date, and Payment schedule. The "Purchase date" for the active row is 04/03/2025.

No need to activate the full account, unless you want to use any specific services.

Free trial status: \$300.00 credit and 88 days remaining. Activate your full account to get unlimited access to all of Google Cloud—use any remaining credits, then pay only for what you use.

Google Cloud

Your orders for this product

Select a billing account * — My Billing Account

This page contains all your orders for the Databricks product. To manage other product orders visit [Your orders](#).

Filter Filter by column name or chart value

Status	Order number	Order title	Provider	Product	Plan
Active ⓘ	[REDACTED]	Databricks	Databricks	Databricks	[REDACTED]

Activate your full account

- Keep your cloud running uninterrupted
- Keep any remaining credits to spend during your Free Trial
- Pay only for what you use—billing starts once your Free Trial ends

Cancel [Activate](#)

Google Cloud My First Project

Product details

Databricks

All your data, analytics and AI on one Lakehouse platform

Trial Active

[MANAGE ON PROVIDER](#) [CONTACT SALES](#) [Last purchased on 4/3/25](#)

OVERVIEW PRICING DOCUMENTATION SUPPORT RELATED PRODUCTS

Overview
Powered by Delta Lake, Databricks combines the best of data warehouses and data lakes into a lakehouse architecture, giving you one platform to collaborate on all of your data, analytics and AI workloads.

Notebooks: Build data science, data engineering and machine learning notebooks using Python, SQL, R, Scala. Collaborate on these notebooks with

Additional details
Type: [SaaS & APIs](#)
Last product update: 6/26/24
Category: [Analytics](#), [Big data](#), [Machine learning](#)

The screenshot shows the Google Cloud Product details page for Databricks. At the top, there's a navigation bar with 'Google Cloud' and 'My First Project'. Below it, a back arrow and the text 'Product details'. The main content area features the Databricks logo and the tagline 'All your data, analytics and AI on one Lakehouse platform'. A green badge indicates 'Trial Active'. Below this, there are tabs for 'OVERVIEW', 'PRICING', 'DOCUMENTATION', 'SUPPORT', and 'RELATED PRODUCTS'. A 'CONTACT SALES' button and a note 'Last purchased on 4/3/25' are also present. A modal window titled 'You're leaving Google' is displayed, stating 'This link opens a partner site. Google isn't responsible for privacy or security on third party sites.' It includes 'CANCEL' and 'OK' buttons. The 'OVERVIEW' tab is selected, showing a brief description of Databricks and its capabilities.

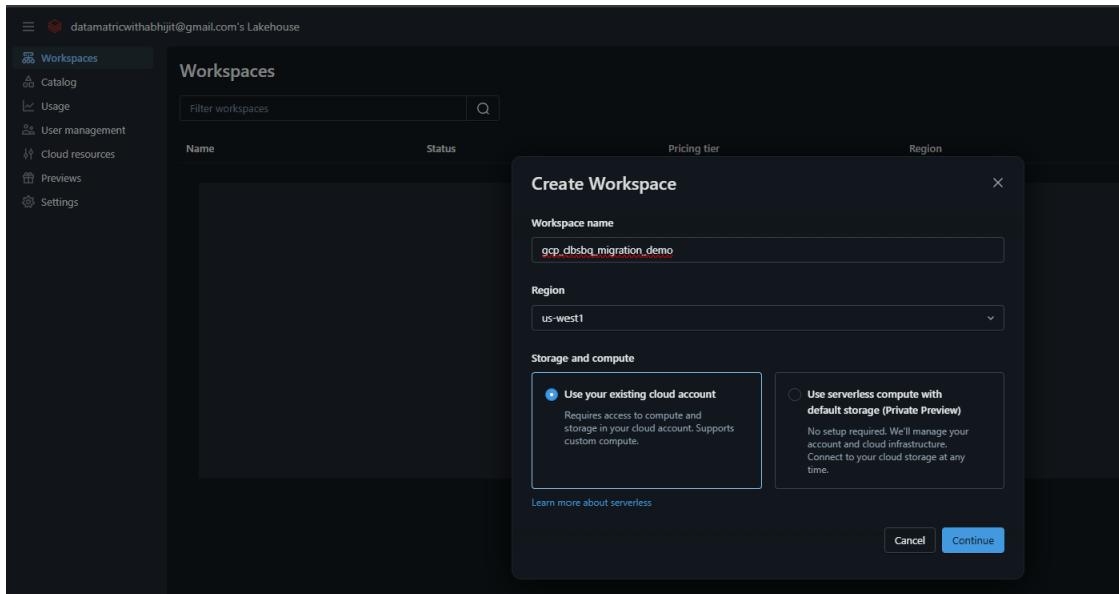
Click on “Continue With Google” account that we used for Databricks account registration and select the plan

The screenshot shows the 'Select a subscription plan' page from accounts.gcp.databricks.com. The URL in the address bar is https://accounts.gcp.databricks.com/subscription-plan-select?account_id=... . The page has a dark background. At the top, it says 'Select a subscription plan' with a 'Most popular' button. Below it is a 'Premium' section with a list of features: 'Cloud native security and autoscaling', 'Databricks SQL', 'IP access list', and 'Role based access control'. The 'Role based access control' item has a checked checkbox and a 'Selected' label. A note at the bottom states 'Your 14-day free trial starts when you click Continue. Thereafter, you will be charged at the list rates.' A large blue 'Continue' button is centered at the bottom. Below the button, there are terms and conditions: 'By clicking continue you agree to Databricks terms and conditions.' and 'If your company has an existing contract with Databricks, please talk to your company contact before creating a Databricks environment so that any negotiated discounts are applied.'

The screenshot shows the Databricks Workspaces page. On the left, there's a sidebar with options like 'Workspaces', 'Catalog', 'Usage', 'User management', 'Cloud resources', 'Previews', and 'Settings'. The main area has a heading 'Workspaces' with a search bar. Below it is a table with columns 'Name', 'Status', 'Pricing tier', 'Region', and 'Created'. A large central placeholder image features a stylized 'D' shape with arrows pointing in and out, labeled 'Workspaces'. Text below the image says 'Your workspace is the environment for doing work in Databricks. Create one to get started.' and a 'Create workspace' button.

To create workspace, we need to fetch/copy GCP project-id.

The screenshot shows the Google Cloud Platform interface. At the top, there's a navigation bar with 'Google Cloud' and 'GCP Databricks Project'. A search bar is also at the top. A 'Welcome Abhiit Das' message is displayed. A 'Select a project' dialog box is open in the center. It shows a list of recent projects: 'GCP Databricks Project' (selected) and 'My First Project'. There's a search bar and tabs for 'RECENT', 'STARRED', and 'ALL'. A 'NEW PROJECT' button is at the top right of the dialog. The background shows 'Recommend' and 'Pre-built solutions' sections.



The screenshot shows the Databricks Workspaces interface after the workspace has been created. The workspace 'gcp_dbbsq_migration_demo' is listed in the workspace list with the following details: Name 'gcp_dbbsq_migration_demo', Status 'Provisioning', Pricing tier 'Premium', and Region 'us-west1'.

Create Databricks cluster - >

Check CPU availability

The screenshot shows the Google Cloud IAM & Admin / Quotas page for the project "GCP-DBSBQ-Migration". The left sidebar shows navigation links for IAM, PAM, Principal Access Boundaries, Organizations, Identity & Organization, Policy Troubleshooter, Policy Analyzer, Organization Policies, Service Accounts, Workload Identity Federation, Workforce Identity Federation, Labels, and Tags. The main content area displays a table titled "Quotas & System Limits for project 'GCP-DBSBQ-Migration'". The table lists eight entries for the Compute Engine API, each showing a quota of N2 CPUs (8), a region (us-east1, us-east4, us-east5, us-south1, us-west1, us-west2, us-west3, us-west4), and a status of 0% used. There is a 'MANAGE ALERT POLICIES' button at the top right of the table.

Now Databricks workspace is created. Next we will create a service account to access GCS bucket from Databricks.

The screenshot shows the Google Cloud IAM & Admin / Service accounts page. The left sidebar is collapsed. The main area displays a table of service accounts for the project "████████migration".

Email	Status	Name	Description	Key ID	Key creation date	OAuth 2 Client ID	Actions
[REDACTED]	Enabled	Compute Engine default service account	No keys	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	Enabled	Databricks Compute Service Account	Default service account attached to VMs for Databricks clusters with no custom service account	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
gcp-dbs-gcs-bq-demo@gcp-dbsbq-migration.iam.gserviceaccount.com	Enabled	gcp-databricks-gcs	No keys	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]

Provide necessary permission to the service account in GCS bucket

The screenshot shows the Google Cloud Cloud Storage Bucket details page for the "retail-demo-data-abhijit" bucket. The left sidebar shows "Cloud Storage" selected under "Buckets". The main area has "Permissions" selected. A "Grant access" section is open, showing a table of principals and their roles.

Type	Principal	Name	Role
[REDACTED]	[REDACTED]	Compute Engine default service account	Storage Object Viewer
[REDACTED]	[REDACTED]	Legacy Cloud Build Service Account	Cloud Build Service Account
[REDACTED]	[REDACTED]	[REDACTED]	Databricks Resource Role V2
[REDACTED]	[REDACTED]	[REDACTED]	Databricks Service IAM Role
[REDACTED]	[REDACTED]	[REDACTED]	Storage Insights Collector Service
[REDACTED]	[REDACTED]	[REDACTED]	Storage Object Admin
[REDACTED]	[REDACTED]	[REDACTED]	Storage Legacy Bucket Owner
[REDACTED]	[REDACTED]	[REDACTED]	Storage Legacy Object Owner

On the right, there are sections for "Grant access to 'retail-demo-data-abhijit'" and "Add principals". A new principal "no@gcp-dbsbq-migration.iam.gserviceaccount.com" is listed. The "Assign roles" section shows a role dropdown set to "Storage Admin" and a note about grants full control of buckets and objects.

Mount GCS buckets to access retails source data from GCP to Databricks. And create bronze, silver and gold databases in the Databricks default unity catalog.

Environment Setup Python Tabs: OFF ★
File Edit View Run Help Last edit was 3 minutes ago

GCS Databricks Mount

```
▶ ✓ 09:51 PM (12s) 2
bucket_name = "████████_data-abhijit"
mount_name = "gcsmount_████"
dbutils.fs.mount(
    source = f"gs://{bucket_name}",
    mount_point = f"/mnt/databricks/{mount_name}",
    extra_configs = {"fs.gs.project.id": "████████_migration"}
)
True
```

```
▶ ✓ 3 minutes ago (1s) 3
df_csv = spark.read.format("csv").option("header", "true").load("/mnt/databricks ...")
```

```
▶ ✓ 3 minutes ago (2s) 4
# Create Bronze Layer Database
spark.sql("CREATE DATABASE IF NOT EXISTS bronze_db")

# Create Silver Layer Database
spark.sql("CREATE DATABASE IF NOT EXISTS silver_db")

# Create Gold Layer Database
spark.sql("CREATE DATABASE IF NOT EXISTS gold_db")
DataFrame[]
```

Agentic AI Use Case:

```
server_hostname="dbc-xxxxx.cloud.databricks.com", # Replace with your workspace hostname
http_path="/sql/1.0/warehouses/xxxxxx",      # Replace with your SQL warehouse path
access_token="dapi-xxxxxxxxxxxxxxxxxxxx"       # Use personal access token or notebook-scoped
```

Create PAT:

The screenshot shows the Databricks Settings interface under the 'Access tokens' section. A modal window titled 'Generate new token' is open, displaying a successfully created token: 'dapi/776533a07455b6577093d051'. A warning message says, 'Make sure to copy the token now. You won't be able to see it again.' There is a 'Done' button at the bottom right of the modal.

Get from SQL warehouse path:

The screenshot shows the Databricks SQL Warehouses interface under the 'Connection details' tab for the 'Serverless Starter Warehouse'. It displays connection details for various tools: Tableau, Power BI, dbt, Python, Java, Node.js, Go, and More tools. Below these are fields for 'Server hostname' (containing 'https://2356902083804511.gcp.databricks.com') and 'HTTP path' (containing '/sql/1.0/warehouses').

Create notebooks:

1. BronzeDataProcess

File Edit View Run Help Python Tabs: ON Last edit was 15 days ago Run all Terminated Schedule (1)

Raw data load to Bronze Layer - Bronze Layer Data Processing

```

▶ Aug 03, 2025 (1g)
%run /Workspace/Users/datametricwithabhijit@gmail.com/util_mod

folders = [
    # '/mnt/databricks/gcsmount_ecommerce/category_name',
    'dbfs:/mnt/databricks/gcsmount_retails/customers/customers.csv',
    'dbfs:/mnt/databricks/gcsmount_retails/inventory/inventory.csv',
    'dbfs:/mnt/databricks/gcsmount_retails/order_items/order_items.csv',
    'dbfs:/mnt/databricks/gcsmount_retails/orders/orders.csv',
    'dbfs:/mnt/databricks/gcsmount_retails/products/products.csv',
    'dbfs:/mnt/databricks/gcsmount_retails/stores/stores.csv',
]

load_raw_data(folders)

# (10) Spark Jobs
Data loaded successfully for customers in bronze_db
Data loaded successfully for inventory in bronze_db
Data loaded successfully for order_items in bronze_db
Data loaded successfully for orders in bronze_db
Data loaded successfully for products in bronze_db
Data loaded successfully for stores in bronze_db
All data loaded successfully

dbutils.notebook.exit("SUCCESS")

```

1. BronzeDataProcess 2. SilverDataProcess

File Edit View Run Help Python Tabs: ON Last edit was 14 days ago Run all Terminated Schedule (1)

Bronze To Silver - Silver Layer Data Processing

```

▶ Aug 03, 2025 (26)
# Create orders
df_orders = spark.sql("""
SELECT o.order_id,
       c.customer_id,
       c.first_name,
       c.last_name,
       o.store_id,
       s.store_name,
       o.order_date,
       o.status
FROM gcp_dbsbq_migration_demo.bronze_db.orders o
JOIN gcp_dbsbq_migration_demo.bronze_db.customers c USING (customer_id)
JOIN gcp_dbsbq_migration_demo.bronze_db.stores s USING (store_id)
""")
df_orders.write.format("delta").mode("overwrite").saveAsTable("gcp_dbsbq_migration_demo.silver_db.orders")

# (7) Spark Jobs
df_orders: pyspark.sql.dataframe.DataFrame = [order_id: string, customer_id: string ... 6 more fields]

▶ Aug 03, 2025 (55)
# Create order_items
df_order_items = spark.sql("""
SELECT oi.order_id,
       oi.product_id,
       p.name AS product_name,
       p.category,
       oi.quantity,
       oi.unit_price,
       oi.total price

```

Silver To Gold - Gold Layer Data Processing

```

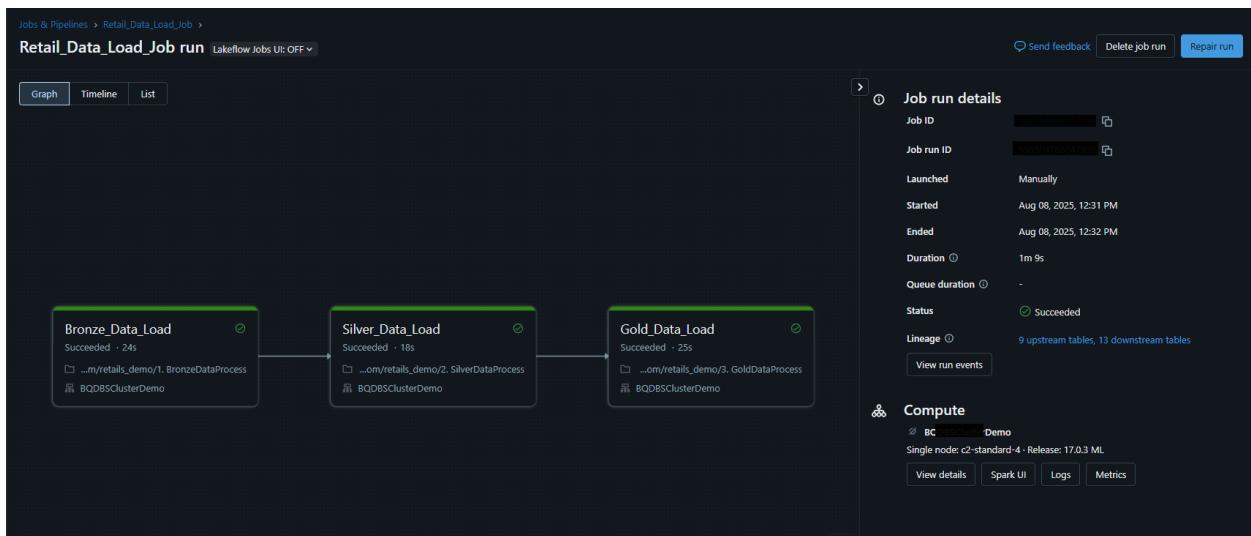
2 Aug 03, 2025 (5s)
# Create gold_daily_store_sales
df_gold_daily_store_sales = spark.sql("""
SELECT DATE(o.order_date) AS order_date,
o.store_id,
o.store_name,
COUNT(DISTINCT o.order_id) AS total_orders,
SUM(o.total_price) AS revenue
FROM gcp_dbsq_migration_demo.silver_db.orders o
JOIN gcp_dbsq_migration_demo.silver_db.order_items oi USING (order_id)
WHERE o.status = 'completed'
GROUP BY 1, 2, 3
""")
df_gold_daily_store_sales.write.format("delta").mode("overwrite").saveAsTable("gcp_dbsq_migration_demo.gold_db.gold_daily_store_sales")

(7) Spark Jobs
df_gold_daily_store_sales: pyspark.sql.dataframe.DataFrame = [order_date: date, store_id: string ... 3 more fields]

3 Aug 03, 2025 (3s)
# Create gold_top_products
df_gold_top_products = spark.sql("""
SELECT product_id,
product_name,
category,
SUM(quantity) AS total_quantity_sold,
SUM(total_price) AS total_revenue
FROM gcp_dbsq_migration_demo.silver_db.order_items
GROUP BY 1, 2, 3
ORDER BY total_revenue DESC
LIMIT 20
""")

```

Workflow:



Lakebridge – Databricks Lab Accelerator:

+ MCP Demo X 3. Agentic Rag Demo Lakebridge In Databricks BigQueryToDatabricksConversion Remorph in Databricks 4. LangChain Basic Setup and RAG Demo 5. Llamaindex Basic RAG Demo

File Edit View Run Help Python Tabs: ON Last edit was 32 minutes ago

Lakebridge in Databricks

Lakebridge is an open-source toolkit developed by Databricks Labs to simplify and accelerate data platform migration to the Databricks Lakehouse. It supports transitioning from systems such as Snowflake, Oracle, SQL Server, Hive, and Presto.

Key Capabilities

- Analyze: Profiles SQL workloads to assess readiness and complexity for migration.
- Convert: Translates SQL scripts from source dialects (e.g., Snowflake, Oracle) into Databricks SQL.
- Validate: Compares data between source and Databricks to ensure correctness after migration.

Usage

- Lakebridge is executed via the Databricks Labs CLI.
- Requires YAML configuration files.
- Can be run from:
 - Databricks Web Terminal (on supported clusters)
 - Local machines with Databricks CLI access

Note: Lakebridge is not designed for direct use within notebooks. It does not expose public Python APIs for in-notebook execution.

Resources

- GitHub: github.com/databrickslabs/lakebridge
- Blog: [Databricks Lakebridge Announcement](#)

+ MCP Demo 3. Agentic Rag Demo Lakebridge In Databricks BigQueryToDatabricksConversion Remorph in Databricks 4. LangChain Basic Setup and RAG Demo

File Edit View Run Help Python Tabs: ON Last edit was 10 days ago

```

# df_csv = spark.read.format("csv").option("header", "true").load("/mnt/databricks/gcsmount_retail/stores/stores.csv")

def convert_bq_to_delta(query: str) -> str:
    output = []

    # Map old database names to new ones
    db_mapping = {
        "gcp-dbsdq-migration.retail_bronze": "gcp_dbsdq_migration_demo.bronze_db",
        "gcp-dbsdq-migration.retail_silver": "gcp_dbsdq_migration_demo.silver_db",
        "gcp-dbsdq-migration.retail_gold": "gcp_dbsdq_migration_demo.gold_db"
    }

    # Replace all schema names based on mapping
    for old_db, new_db in db_mapping.items():
        query = query.replace(old_db, new_db)

    # Step 1: Convert CREATE SCHEMA to spark.sql
    schema_matches = re.findall(r"CREATE SCHEMA IF NOT EXISTS `(.+?)`;", query)
    for schema in schema_matches:
        output.append(f"spark.sql(\"CREATE DATABASE IF NOT EXISTS `{schema}`\")")

    # Step 2: Extract CREATE TABLE AS SELECT blocks
    statements = re.findall(
        r"CREATE OR REPLACE TABLE (.+?) AS\s+SELECT(.+?);(?=|\s*CREATE|\s*$)", query,
        re.DOTALL
    )

    for full_table_name, select_block in statements:
        full_table_name = full_table_name.strip()
        table_short_name = full_table_name.split(".")[1]
        output.append(f"\nCreate {table_short_name}")
        output.append(f"df_{table_short_name} = spark.sql(\"\"\"{select_block.strip()}\"\"\")")
        output.append(f"df_{table_short_name}.write.format(\"delta\").mode(\"overwrite\").saveAsTable(\"{full_table_name}\")\n")

    return "\n".join(output)

```

```

LIMIT 20
""")
df_gold_top_products.write.format("delta").mode("overwrite").saveAsTable("gcp_dbsbq_migration_demo.gold_db.gold_top_products")

```

Silver Query Conversion

```

▶ Aug 08, 2025 (c1s) 8
bq_query_4 = """ --orders with proper dates & joins CREATE OR REPLACE TABLE gcp- ...

```

```

▶ Aug 08, 2025 (2s) 9
# Create orders
df_orders = spark.sql("""
SELECT o.order_id,
       o.customer_id,
       c.first_name,
       c.last_name,
       o.store_id,
       s.store_name,
       o.order_date,
       o.status
FROM gcp_dbsbq_migration_demo.bronze_db.orders o
JOIN gcp_dbsbq_migration_demo.bronze_db.customers c USING (customer_id)
JOIN gcp_dbsbq_migration_demo.bronze_db.stores s USING (store_id)
""")
df_orders.write.format("delta").mode("overwrite").saveAsTable("gcp_dbsbq_migration_demo.silver_db.orders")

```

(3) Spark Jobs

df_orders: pyspark.sql.dataframe.DataFrame = [order_id: string, customer_id: string ... 6 more fields]

Remorph: Databricks Labs Migration Toolkit

Remorph is an open-source toolkit developed by [Databricks Labs](#) to simplify and accelerate migration and onboarding to Databricks. It offers two core capabilities:

1. Transpile

A SQL code converter that automatically translates SQL scripts from other dialects—especially [Snowflake](#)—into [Databricks SQL](#), using the [SQLGlot](#) parser and built-in validation.

2. Reconcile

A data reconciliation tool that compares data between a [source system](#) (like [Snowflake](#) or [Oracle](#)) and [Databricks](#), identifying mismatches in data or schema to ensure migration accuracy.

Use Remorph to speed up SQL code conversion, validate transformations, and ensure data consistency during platform migrations.

```

▶ Aug 08, 2025 (10s) 2
!pip install databricks-labs-remorph

```

```

▶ Aug 08, 2025 (4s) 3
dbutils.library.restartPython()

```

```

▶ Aug 08, 2025 (c1s) 4
import sys print(sys.version)

```

```

▶ Aug 08, 2025 (2s) 5

```

```

5
6
%pip install sqlglot
Requirement already satisfied: sqlglot in /local_disk0/.ephemeral_nfs/envs/pythonEnv-9229850d-f0bd-4a2f-b283-4e1636b88ecd/lib/python3.12/site-packages (25.32.1)
Note: you may need to restart the kernel using %restart_python or dbutils.library.restartPython() to use updated packages.

7
dbutils.library.restartPython()

8
import sqlglot

converted = sqlglot.transpile(
    "SELECT DATE_DIFF(CURRENT_DATE(), DATE(order_date), DAY) AS days_since_order FROM orders",
    read="bigquery",
    write="databricks"
)

print(converted[0])

SELECT DATEDIFF(DAY, DATE(order_date), CURRENT_DATE) AS days_since_order FROM orders

```

Agentic AI Introduction:

Workspace > Users > abhijit.das@gmail.com >

AI_ML_Solutions ★

Name	Type	Owner	Created at
.gradle	Folder	Abhijit Das	Aug 04, 2025, 12:16 AM
lc_demo	Folder	Abhijit Das	Aug 07, 2025, 01:01 PM
rag_chroma	Folder	Abhijit Das	Aug 05, 2025, 11:37 AM
rag_mcp_demo	Folder	Abhijit Das	Aug 05, 2025, 12:04 PM
1. Agentic AI Demo	Notebook	Abhijit Das	Aug 03, 2025, 11:50 PM
2. Agentic Rag + MCP Demo	Notebook	Abhijit Das	Aug 05, 2025, 12:01 PM
3. Agentic Rag Demo	Notebook	Abhijit Das	Aug 05, 2025, 11:29 AM
4. LangChain Basic Setup and RAG Demo	Notebook	Abhijit Das	Aug 07, 2025, 12:56 PM
5. Llamaindex Basic RAG Demo	Notebook	Abhijit Das	Aug 07, 2025, 01:05 PM
6. Haystack Basic RAG Demo	Notebook	Abhijit Das	Aug 07, 2025, 01:10 PM
7. OpenLLM by BentoML – Code Generator Demo	Notebook	Abhijit Das	Aug 07, 2025, 01:24 PM
DL: Customer Lifetime Value (CLV) Tier Prediction	Notebook	Abhijit Das	Aug 07, 2025, 01:47 PM
Generate Synthetic Customer Data - Using GAN	Notebook	Abhijit Das	Aug 07, 2025, 01:59 PM
sample_doc.txt	File	Abhijit Das	Aug 07, 2025, 01:07 PM
Untitled Notebook 2025-08-12 09:07:43	Notebook	Abhijit Das	Aug 12, 2025, 09:07 AM

1. BronzeDataProcess 1. Agentic AI Demo +

File Edit View Run Help Python Tabs ON Last edit was 10 days ago

Agentic AI Demo

This notebook demonstrates how to build an **Agentic AI-powered data assistant** that:

- Accepts business questions in **natural language**
- Translates them into **SQL queries**
- Executes the SQL on **Databricks SQL Warehouse**
- Displays results in a friendly **Gradio-based UI**

Key Components

1. Library Installation

Installs essential packages:

- `langchain`, `openai`, `databricks-sql-connector` – for LLM + Databricks integration
- `gradio` – for UI interaction

2. Databricks SQL Connection

Defines a utility function to connect to Databricks SQL using credentials stored in `DATABRICKS_CONFIG`. This is used to run SQL queries generated by the AI.

3. LLM SQL Generator (LangChain + GPT-4)

- Uses `LangChain` and `ChatOpenAI (gpt-4)`
- Prompt is tailored to generate SQL against a known schema in `gcp_dbsq_migration_demo.gold_db`
- Tables used:
 - `gold_customer_features`
 - `gold_daily_store_sales`

2. Agentic Rag + MCP Demo 1. Agentic AI Demo +

File Edit View Run Help Python Tabs ON Last edit was 10 days ago

Aug 08, 2025 (7s) 13

```

    """
    <h1 style="color:#0066cc; text-align:center;">💡 Agentic AI: Retail Sales Assistant</h1>
    <p style="text-align:center; font-size: 16px;">
        Ask questions in plain English about store <b>revenue, orders, or performance</b>.<br>
        Example: <i>"What is the total number of orders?"</i>
    </p>
    """
)

with gr.Row():
    with gr.Column(scale=1):
        user_input = gr.Textbox(
            label="Ask a Business Question",
            placeholder="e.g. What was the total revenue last month?",
            lines=2,
            show_label=True
        )
        submit_btn = gr.Button("👉 Submit")

    with gr.Column(scale=2):
        sql_output = gr.Markdown(label="🔮 SQL Generated")
        table_output = gr.DataFrame(label="📊 Result Table", wrap=True)

    submit_btn.click(
        fn=agentic_ai,
        inputs=user_input,
        outputs=[sql_output, table_output]
    )

demo.launch(share=True)

* Running on local URL: http://127.0.0.1:8000
* Running on public URL: https://gradio.live

```

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working directory to deploy to Hugging Face Spaces (<https://huggingface.co>)

Agentic AI + RAG + MCP

Agentic AI Assistant with RAG + Databricks SQL

This notebook builds a simple yet powerful **Agentic AI** assistant capable of:

- Understanding natural language queries
- Retrieving relevant schema context using RAG
- Generating SQL queries via GPT-4
- Executing the SQL on Databricks
- Returning clean, readable results

Step 1: Install Required Libraries

Install all necessary Python libraries such as:

- LangChain
- OpenAI
- ChromaDB
- Databricks SQL Connector

Step 2: Restart Python Kernel

Restart the Python environment to ensure all installed packages are correctly loaded.

Step 3: Import Utility Configuration

Run a helper notebook (`ai_util_mod`) to load any Databricks-specific configs like `DATABRICKS_CONFIG`.

Step 4: Load Schema and Create Vector Store

```
# Ask Agentic RAG Question
response = agent_executor.invoke({
    "input": "Share the top 10 products",
    "chat_history": [],
    "agent_scratchpad": ""
})
print(response["output"])

sql
SELECT *
FROM gcp_dbsq_migration_demo.gold_db.gold_top_products
ORDER BY total_quantity_sold DESC
LIMIT 10;
```
[3] SQL Result

product_id	product_name	category	total_quantity_sold	total_revenue
510b3cab-3715-4d44-9c9e-8b195849b3e	Trouble	Clothing	119	54383
13110f93-d3c0-4dc4-ad23-d608c9c3f6ff	Consumer	Toys	87	32966.8
29211965-577a-4299-a222-13601fc647ea	Apply	Electronics	82	35302.6
a2c3672b-eadc-4137-b29c-6be09a42cf0	Investment	Toys	82	38887.8
e4c4dd191-448c-4df4-9547-62a3ad4553eb	Reveal	Electronics	79	31876.5
ed96866b-b3a2-4440-be54-82313344de8d	Student	Home	78	27487.2
94a62682-3bc8-4beb-bc73-daf875eab459	Outside	Toys	74	32326.9
f115d6b1-2876-4cb5-927e-710e1ef8b5eF	Eye	Clothing	72	28324.1
225609a0-0fb0-4e77-9d58-ed3e690b731d	International	Clothing	71	31212.3
e212286f-4991-48cc-8399-dc53d103d47a	Statement	Books	69	28216.2

MLflow Trace UI Learn More
Send feedback
```

Agentic AI + RAG

**Agentic RAG SQL Assistant in Databricks – Step-by-Step Summary**

This notebook builds a lightweight Retrieval-Augmented Generation (RAG) assistant using **LangChain GPT-4**, and **Databricks SQL** to answer business questions with context-aware SQL generation and execution.

### Step 1: Install Required Libraries

Install essential libraries:

- openai, langchain, tiktoken, chromadb, databricks-sql-connector

These enable interaction with LLMs, vector search, and Databricks SQL.

### Step 2: Restart Python Kernel

Restart the Python environment to ensure proper loading of newly installed packages.

### Step 3: Import Databricks Config

Run a utility notebook ([ai\\_util\\_mod](#)) to import `DATABRICKS_CONFIG` used for SQL connections.

### Step 4: Create SQL Execution Function

Defines a utility function `run_sql(sql)` to:

- Connect to Databricks SQL
- Run the generated SQL query
- Return results as a Pandas DataFrame
- Handle any execution errors

```
sql_chain = LLMChain(llm=llm, prompt=sql_prompt)

#Test the Agentic RAG

question = "Share top 10 products"
sql, result_df = agentic_rag_sql(question)

print("Generated SQL:\n", sql)
display(result_df)

result_df: pandas.core.frame.DataFrame = [product_id:object, product_name: object ... 2 more fields]

SELECT product_id, product_name, total_quantity_sold, total_revenue
FROM gcp_dbsq_migration_demo.gold_db.gold_top_products
ORDER BY total_quantity_sold DESC
LIMIT 10;
```

| product_id                            | product_name  | total_quantity_sold | total_revenue      |
|---------------------------------------|---------------|---------------------|--------------------|
| 510b3ca0-3715-4d44-9c9a-8b195849eb3e  | Trouble       | 119                 | 54383              |
| 1311093-d3c0-4ddc-ad23-d60b9c036ff    | Consumer      | 87                  | 32960.820000000002 |
| 2921196-577a-4299-a222-13601fc647ae   | Apply         | 82                  | 35302.640000000014 |
| a2c3672b-eddc-4137-b29c-0be09a42cdff  | Investment    | 82                  | 30887.760000000017 |
| ec49d191-448c-4d44-9547-6283ad4553eb  | Reveal        | 79                  | 31876.5            |
| ed96866b-b3a2-4x40-be54-823133d4de... | Student       | 78                  | 27487.2            |
| 94a62082-3bc0-4beb-bc73-daf875eb45b   | Outside       | 74                  | 32326.899999999994 |
| f115d6b1-2076-4cb5-927e-710e1ef885ef  | Eye           | 72                  | 28324.079999999987 |
| 225609a0-0fb4-e77-9d58-ed3e690b731d   | International | 71                  | 31212.310000000002 |

